10th International Command and Control Research and Technology Symposium
The Future of C2
June 13-16, 2005, McLean, VA

The Generalized Weapon Target Assignment Problem

Jay M. Rosenberger
Hee Su Hwang
Ratna P. Pallerla
Department of Industrial and
Manufacturing Systems Engineering
The University of Texas at Arlington
P. O. Box 19017
Arlington, Texas 76019

Adnan Yucel
Ron L. Wilson
Ed G. Brungardt
Lockheed Martin Aeronautics Company
MZ 9334
P. O. Box 748
Fort Worth, Texas 76101

Point of Contact

Adnan Yucel
Tel: 817/935-1342
Fax: 817/777-2115
Email: Adnan.Yucel@lmco.com

Classification: UNCLASSIFIED

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **The Generalized Weapon Target Assignment Problem**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Lockheed Martin Aeronautics Company,MZ 9334,PO Box 748,Fort Worth,TX,76101**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
10th International Command and Control Research and Technology Symposium

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**33**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Abstract

Dynamic command and control and battle management functions require fast and effective decision aids to provide optimal allocation of resources (object/sensor pairing, weapon/target assignment) for effective engagement and real-time battle damage assessment. The basic Weapon Target Assignment (WTA) problem considers the assignment of a set of platforms/weapons to a set of targets such that the overall expected effect is maximized. In the present study, we extend the basic WTA problem by allowing for multiple target assignments per platform, subject to the number of weapons available and their effectiveness. We formulate the problem as a linear integer programming problem and investigate two solution methods. The first method is a greedy approach based on the sequential application of the auction algorithm that was generalized for assigning *n* assets/resources to *m* targets. The second method is built on a branch-and-bound framework that enumerates feasible tours of assets/resources – a process that can become computationally intensive with increasing number of sources and targets but will find an optimal solution. We provide results of Monte Carlo experiments and provide comparative evaluation of the two solution methods. Finally, we extend the brand-and-bound technique to assigning multiple platforms per target and thereby demonstrate its utility for collaborative asset planning. While this study focuses on weapon target pairing for illustration purposes, the methods and results herein are readily applicable to sensor tasking and similar resource allocation problems.

1. Introduction

A key component in planning and dynamic control of missions is the assignment of resources (e.g., different aircraft types and weapons) to targets. The Weapon Target Assignment (WTA) problem is to find a proper assignment of platforms/weapons to targets with the objective of maximizing the overall effect associated with targets. Various methods for solving this NP-complete WTA problem have been reported in the literature [1-6]. These studies have focused weapon pairing aspects or sortie analysis with regard to targets. In this study, we consider one or more types of weapons carried by a set of platforms against a set of targets, and extend the basic WTA problem by allowing for multiple target assignments per platform. We also investigate how the formulation can be applied to collaborative planning where multiple sources may be required per target.

2. Model

This section describes the integer programming formulation of the generalized weapon arget assignment problem. Suppose a ground command center or an airborne mission command center has to reassign a set of platforms or reallocate their weapons to a set of targets. Each platform is assumed to carry one or more weapon types, and each target is fully or partially satisfied by one type of weapon served by a platform (referred to as the “source” hereafter). A source is able to serve multiple targets, and we assume it delivers only one type of weapon when it visits a target. A source leaves the starting location, serves different targets and returns to the ending location. The source cannot travel over the distance of its travel capacity (based on available and bingo fuel). An assignment (or task) can be described as assigning a source to targets with proper weapons satisfying the travel capacity limit.

We assume that weapon effectiveness factors (range from 0 to 1) related to the specified quantities of a weapon type and target type are given. Targets are also assigned values to reflect their significance and priority. The benefit of an assignment, which considers each source-target-weapon combination, may be written as

Benefit = Value x (Weapon Effectiveness) x Plength

where Plength = M / (M + distance), and M is a constant

For example, as shown in Figure 1, source 1 starts at position (1,1), serves target 1 at position (3,7) and target 2 at position (6,8), and ends at (2,1).

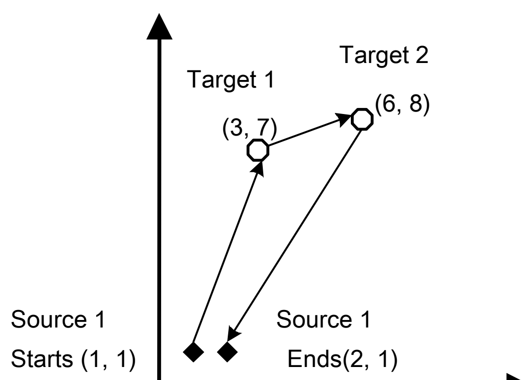


Figure 1. A sample task for source 1

Suppose source 1 has 20 travel capacity units and four weapons of type A, two weapons of type B, and two weapons of type C. Suppose target 1 has a target value of 200 and needs to be served by either one weapon of type A with a combined effectiveness 0.8, one weapon of type B with an effectiveness of 1.0, or two weapons of type C with a combined effectiveness of 0.9. Similarly suppose target 2 has a target value of 100 and needs either two weapons of type A with combined effectiveness 1.0, or one weapon of type B with effectiveness 0.9.

One of possible assignments for source 1 can be starting from (1,1), serving two type C weapons for target 1 and two type A weapons for target 2, and returning (2,1). The benefit from serving target 1 is equal to 179.04 and the benefit from serving target 2 99.68, where M = 1000. Thus the assignment yields total benefit 278.72 with traveling 16.61 distance units.

The objective of the Generalized Weapon Target Assignment Problem is to maximize the total benefit by selecting the best set of assignments for the sources. Suppose that the assignment problem has *m* sources and *n* targets. Then the problem may be formulated as

$$\max \sum_{j \in J} c_j x_j \tag{1}$$

$$s.t. \sum_{j \in Ss} x_j \le 1 \quad s = 1,\ldots,m \tag{2}$$

$$\sum_{j \in Tt} x_j \le 1 \quad t = 1,\ldots,n \tag{3}$$

$$x_j = \begin{cases} 1, & \text{if assignment } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where, J is the set of all feasible assignments and c_j the total benefit from assignment *j*, $j \in J$; S_s, *s* = *1,…,m*, is the subset of assignments to which source *s* is assigned; T_t, *t* = *1,…,n*, is the subset of assignments that serve target *t*. This formulation is similar to the vehicle routing problem.

3. Enumeration of Assignments

This section describes how to enumerate the set of all possible assignments *J*. From target information we can enumerate all target-weapon-quantity sets. For instance, target 1

in the example from Section 2 needs to be served with either one weapon of type A, one weapon of type B, or two weapons of type C. Each one of these weapon and quantity combinations becomes a demand waiting for a proper source, which is called a *target drop*. Therefore, target 1 has three target drops, i.e., target 1-A-1, target 1-B-1 and target 1-C-2. In this manner we can enumerate all possible target drops from target information.

From source and target information we can set all possible assignments, and each of them is composed of a source and sequence of target drops, called a *target drop set*. Each of target drop set can then be combined with a source, called an *assignment*. We can set all possible assignments from source information and target drop sets, and each assignment yields benefit from assigning a source to a target drop set as described in Section 2. Out of these possible assignments we can select feasible assignments, which satisfy the travel capacity limit of related sources.

4. Branch and Bound Algorithm

There are many ways to solve an integer programming problem, and we adapted a simple branch and bound method to handle the weapon target assignment problem because of its flexibility. This section describes the branch and bound algorithm, which is similar to implicit enumeration. Instead of enumerating all possible assignments the algorithm deletes unnecessary enumeration steps and improves its efficiency. Greedy method, skipping unnecessary branching and lower bound rules are used, which will be explained later in this section.

We can see how the branch and bound algorithm works by using enumeration tree in Figure 2. Each node represents a variable, and each branch is a value of a variable. The tree represents four variables and all possible branches. The bottom nodes show all set of solutions. In Figure 2 node 1 implies that X1=1, X2=1, X3=1, and X4=1, and node 2 similarly indicates X1=1, X2=1, X3=1, and X4=0 and so forth. If formulation has four assignment variables, then there are sixteen possible solutions. However, some of them are not feasible because of constraints (2) and (3). Moreover, some solutions are not as good as the others. These properties enable the branch and bound algorithm to delete many of nodes and of branches as shown in Figure 3.

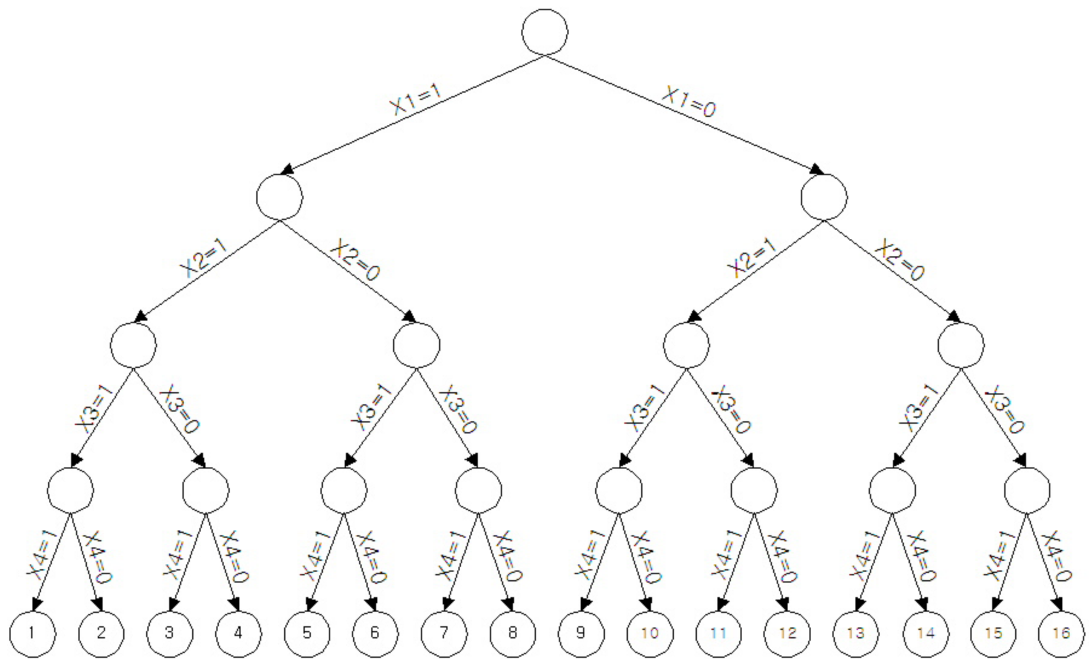


Figure 2. Enumeration Tree

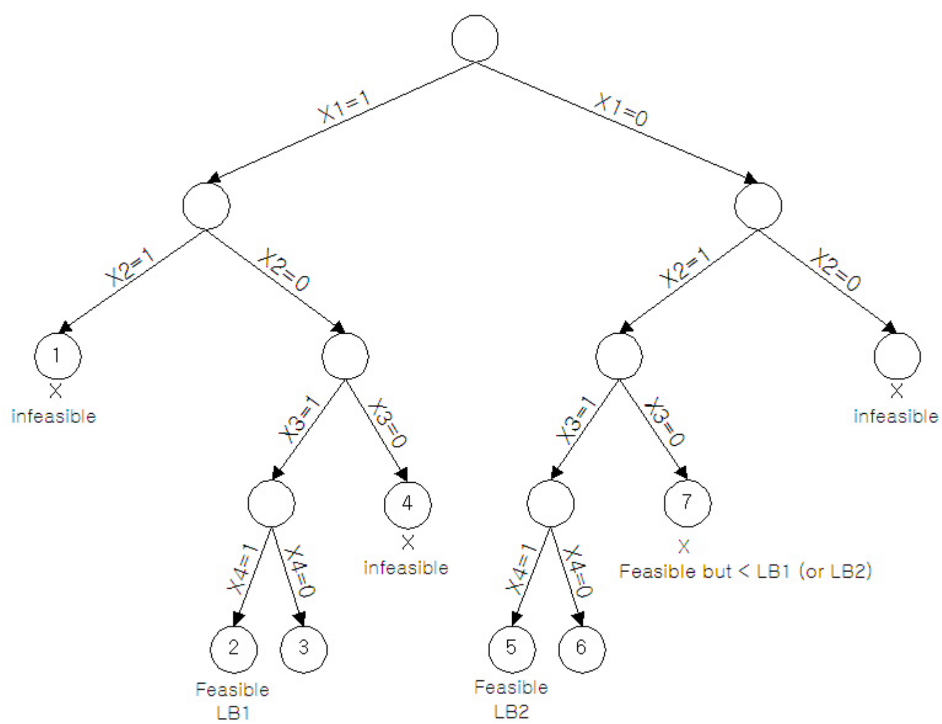


Figure 3. Branch and bound tree

We arbitrarily choose to branch to the left first at each node in Figure 3. Suppose that node 1 is infeasible because of a source conflict constraint (2), so we branch to the right. We can keep branching until we reach node 2 and find a feasible solution. This solution provides a lower bound for other solutions. Because the objective function is nonnegative the solution at node 3 is inferior to the one at node 2. Suppose that node 4 is infeasible because of a constraint; then we need to backtrack until we reach the root node (top node) to branch further.

The algorithm might enumerate all possible assignments. In this case, the total number of nodes is $2^{N+1}-1$, so it would not be practical for large N. The following methods help the algorithm to reduce the number of nodes created by branching recursively and to find (near) optimal solutions as soon as possible. After enumerating all feasible assignments described in Section 3, we sort these assignments in decreasing order of their benefits. The reason for doing this is that we might find a near optimal solution quickly. Suppose the assignments are sorted, and nodes for X1, X2, X3, and X4, in Figure 2, represent the assignment with maximum benefit for the first, second, third, and fourth source, respectively. If node 1 is feasible, then it is optimal because the assignment's benefit is greater than all of assignments for those sources. However, the left-most branch is rarely feasible, so we can select feasible assignments greedily until a feasible solution is found. Because of the greedy selection, the solution is likely near optimal, so we will be able to prune several suboptimal solutions and limit the size of the branch and bound tree.

5. Sequential Method

One heuristic method to solve the generalized weapon target assignment problem is to assign sources to targets sequentially. Given n sources and m targets, we set up a directed bipartite graph (G) of sources and targets, and look for the solution of the asymmetric assignment problem:

$$\max \sum_{(i,j)\in G} a_{ij} x_{ij} \tag{5}$$

$$s.t. \sum_{\{j|(i,j)\in G\}} x_{ij} \le 1 \quad i = 1,\dots,m \tag{6}$$

$$\sum_{\{i|(i,j)\in G\}} x_{ij} \leq 1 \quad j = 1,...,n \tag{7}$$

$$x_{ij} = \begin{cases} 1, & if \ source \ i \ assigned \ to \ target \ j \\ 0, & otherwise \end{cases} \tag{8}$$

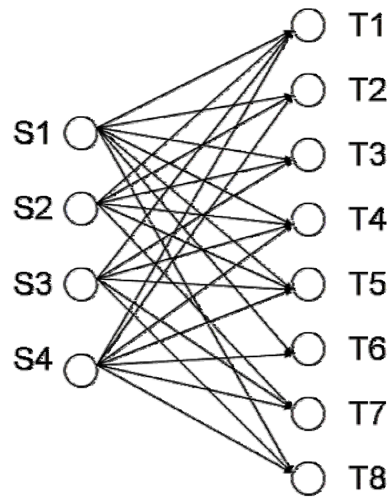


Figure 4. Asymmetric Assignment Problem

In the above, x_{ij} represents a single assignment of source *i* to target *i*, using the resource with the highest effectiveness for that target, and a_{ij} the corresponding benefit for this pairing. (In the example from Section 2, this would be Target Drop 1-B-1, when source 1 is to be assigned to target 1). We use the auction algorithm [7] to solve the above problem. Once the primary assignments are identified, we look for secondary assignments for sources with unallocated resources against any unassigned targets, and keep repeating the bipartite graph build-and-auction process until no feasible assignments remain. This greedy approach is much faster compared to the branch-and-bound scheme, as we enumerate and investigate but only a very small subset of the assignments for the latter (roughly O(nm) vs. O(nm^2)).

7. Extension of branch-and-bound method to multiple source assignments per target

In section 4 we discussed the case where a target is assigned to a single source. Consider the following trivial example, where the assignments are sorted by source, in decreasing benefit order:

Table 1. Single source per target example

Source	**Assignments**
Source 1	X1
Source 2	X2 X3 X4
Source 3	X5 X6

The corresponding enumeration tree is given below.

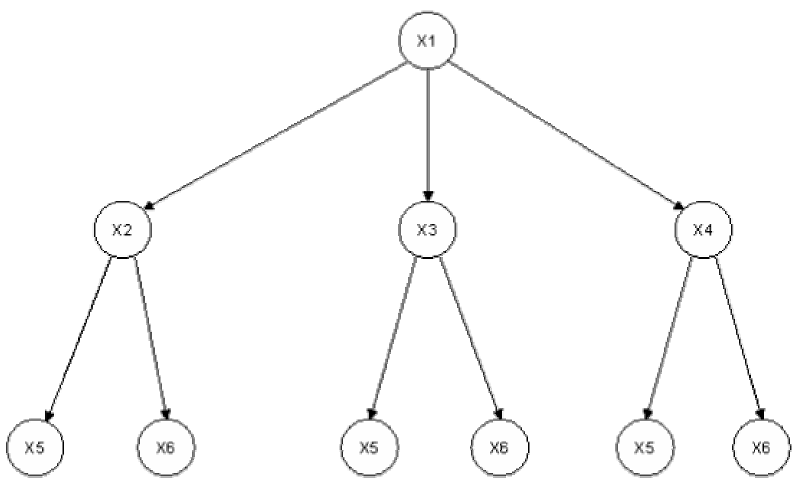


Figure 5. Single source enumeration tree

When we allow more than one source per target, e.g. two platforms each delivering half the total required weapons for a desired effectiveness, an assignment will have pointers two multiple sources. Modifying the example above, where we consider assignment X5 to contain source 2, in addition to source 3, we have

Table 2. Multiple sources per target example

Source	**Assignments**
Source 1	X1
Source 2	X2 X3 X4 X5
Source 3	X5 X6

and the corresponding enumeration tree is shown in Figure 6

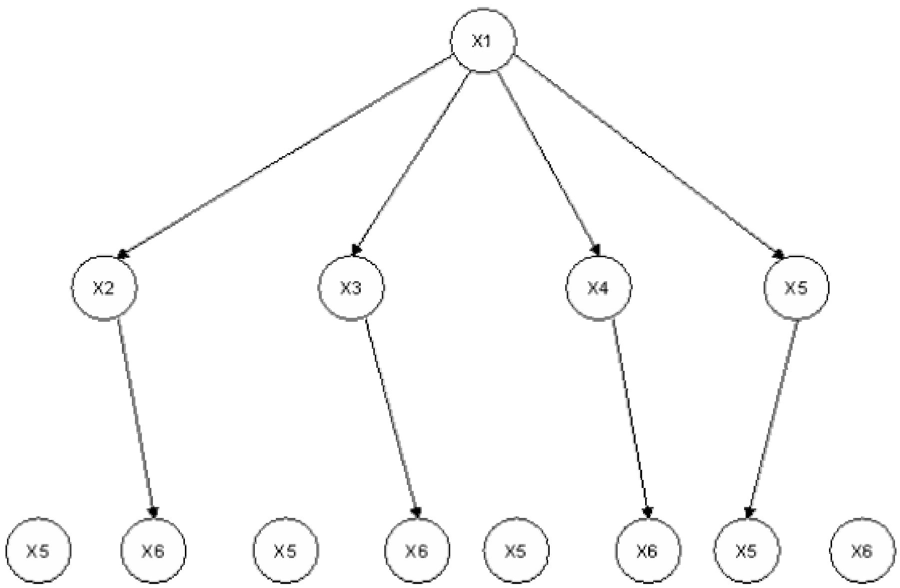


Figure 6. Multiple source enumeration tree

where the branches from X2 to X5, X3 to X5, X4 to X5, and X5 to X6 are automatically eliminated as infeasible branchings. The branch X5 to X5 from source 2 to source three is a fixed branch, dictated by the source constraints.

7. Computational Experiments

We compared the solutions found using the branch-and-bound method with those from the sequential method using random instances from a Monte Carlo Simulation. We then considered the following parameter settings:

Table 3. Base Parameter Set

Variable	**Value**
Max Targets Per Assignment	2
Max Sources Per Assignment	1
Travel Capacity	25
Max Time	10 seconds
Objective Tolerance	0

Max time is the maximum amount of time we allowed the branch-and-bound code to execute, and objective tolerance is used to prune nodes in the branch-and-bound tree that are within a tolerance of the best known feasible solution. By modifying one parameter, we generated fourteen different parameter sets. In each parameter set was used to solve 50 instances randomly generated instances. The average benefit of the corresponding 50 solutions by the branch-and-bound method is compared with that of the sequential method.

Table 4. Results of Monte Carlo Simulation

Max Targets	**Travel Capacity**	**Max Time**	**Max Sources**	**Objective Tolerance**	**Avg BAB Benefit**	**Avg SQM Benefit**	**Avg CPU Time**	**Avg # of Nodes**
2	**25**	**10**	**1**	**0**	**574**	**577**	**5**	**79805**
3	25	10	1	0	**583**	**577**	5	39409
4	25	10	1	0	**585**	**577**	6	29589
2	24	10	1	0	574	577	5	77943
2	23	10	1	0	574	575	5	75986
2	22	10	1	0	574	574	5	66158
2	21	10	1	0	**574**	**571**	4	74098
2	20	10	1	0	**571**	**564**	4	73059
2	25	20	1	0	574	577	9	125654
2	25	30	1	0	574	577	12	168443
2	25	10	2	0	**668**	**577**	5	199158
2	25	10	1	25	573	577	4	70638
2	25	10	1	50	573	577	4	63608
2	25	1	1	0	572	577	1	17580

From Table 4, the best improvement from the sequential method to the branch-and-bound method is in assigning multiple sources to targets. In this case, the average benefit from using the branch-and-bound algorithm is 668, while the average benefit using the sequential method is 577. This is not surprising as the sequential method considers only single source assignments per target. Other significant improvements using the branch-and-bound method in the following parameter sets are summarized below:

Table 5. Sequential vs. B&B method

Change	**Average Sequential Benefit**	**Average B&B Benefit**	**Improvement**
Max Source = 2	577	668	15.8%
Max Targets = 4	577	585	1.38%
Travel Capacity = 20	564	571	1.24%
Max Targets = 3	577	583	1.03%

6. Conclusions

The primary purpose of this study was to investigate the suitability and performance of the successive auction algorithm adopted for a class of multi-target assignment problems. The branch and bound scheme, with its exhaustive search tree, served as the benchmark. Several simulations were performed to compare the sequential method with the branch-and-bound method. We found that the use of successive auctions beyond the primary source-target pairings to generate multi-target assignments produced good results overall. While the branch-and-bound scheme is guaranteed to find optimal results, the successive auction method consistently found multiple assignments that are close to the optimal, with differences that may be considered operationally insignificant. The successive auction method is extremely fast and requires fewer enumerations.

A second objective was to gain insight into the performance of the branch-and-bound method and extend it to problems which may require multiple source assignments per target (or target cluster). In ordering and sorting the feasible assignments by source, and sweeping and pruning feasible solutions by target conflicts and benefit value, we found

that the first feasible solution reached was a good solution with a benefit close to the optimal value (that may be deemed an operationally insignificant difference) in many cases. Thus the branch-and-bound method can be used to provide efficient solutions to the multi-source assignment problem which may be arrived at very quickly through the use of heuristic benefit threshold values and differences.

References

1. Ahuja, R. K., Kumar, A., Jha, K. and Orlin, J. B., "Exact and Heuristic Methods for the Weapon Target Assignment Problem", MIT Sloan Working Paper No. 4464-03, http://ssrn.com/abstract=489802, July 2003.
2. Alexander Toet, A. and de Waard, H., "The Weapon Target Assignment Problem," CALMA Report CALMA.TNO.WP31.AT.95c, February 24, 1994.
3. Kwon, O., Kang, D., Lee, K. and Park, S., "Lagrangian Relaxation Approach to the Targeting Problem," *Naval Research Logistics*, Volume 46, Issue 6, pp. 640-653, 1999.
4. Metler, W.A. and Preston, F.L., "Solutions to a Probabilistic Resource Allocation Problem, " *Decision and Control, Proceedings of the 28th IEEE Conference on* , Dec. 13-15, 1989, Vol. 2, pp. 1606-1611.
5. Lee, Z-J., Su, S-F., and Lee, C-Y., "Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms with Greedy Eugenics," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, Volume: 33, Issue: 1, Feb. 2003, pp. 113-121.
6. Castro, D. R., "Optimization Models for Allocation of Air Strike Assets with Persistence," M.S. Thesis in Operations Research, Naval Postgraduate School, Monterrey, CA, Dec. 2002.
7. Bertsekas, D. P., *Linear Network Optimization: Algorithms and Codes*, MIT Press, Cambridge, Massachusetts, 1991.

Superior Products Through Innovation

Advanced Development Programs

Generalized Weapon Target Assignment Problem

J. Rosenberger, P. Hwang, R. Pallerla, UTA

A. Yucel, R. Wilson, E. Brungardt, LM Aero

presented at

10th ICCRTS, McLean, VA, 14 June 2005

Overview

- Problem Statement
- Branch and Bound Method
- Successive Auction Method
- Computational Experiments
- B & B Extensions
- Summary

Problem - Statement

- **Set of Platforms**
  - ***Resources (type, quantity)***
- **Set of Targets**
- **Assign platforms/resources to targets**
  - ***Assignment: Source {Target/Resource/qty}, Target/Resource/qty},***
  - ***Benefit***
  - ***Target Value***
  - ***Effectiveness (type, quantity)***
- **Goal**
  - ***Maximize overall benefit***

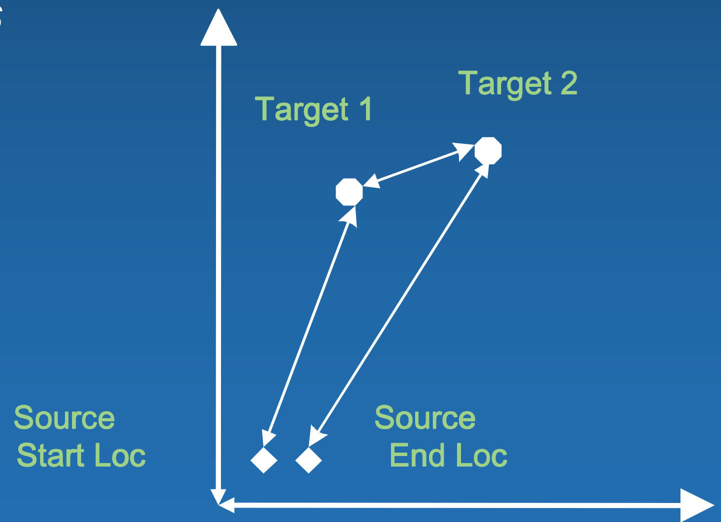
Assignment Problem

- **Basic Assignment Problem**
  - ***m platforms, n targets***
  - ***One target per platform***

- **Generalized Weapon-Target Assignment Problem**
  - ***Multiple target assignments per platform***
  - ***Multiple sources per target***

Enumeration of Assignments

- **Target Drop**
  - *Target/Resource/qty*
  - *Benefit = Target Value * Resource Effectiveness*
- **Target Drop Set**
  - *One or more Target Drops*
  - *Total Benefit = Sum of Target Drop Benefits*
- **Assignment**
  - *Source <–> Target Drop Set*
  - *Benefit = Target Drop Set Benefit * Plength*
  - *Feasible - travel capacity constraint*
  - *Additional considerations*



Problem - Formulation

Formulated as an integer programming problem

$$\max \sum_{j \in J} c_j x_j$$

$$s.t. \sum_{j \in Ss} x_j \le 1 \quad s = 1, \ldots, m$$

$$\sum_{j \in Tt} x_j \le 1 \quad t = 1, \ldots, n$$

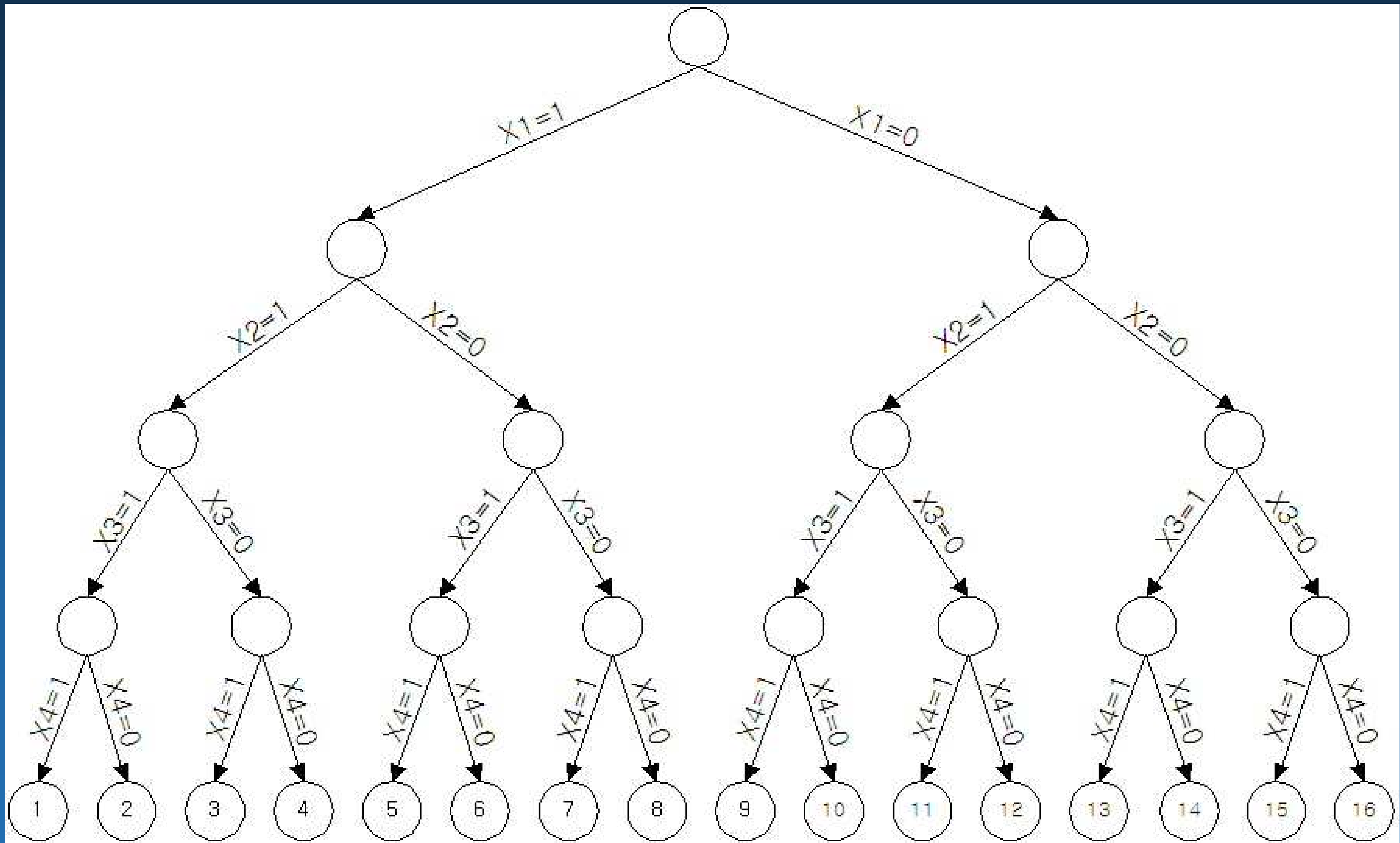
$$x_j = \begin{cases} 1, & \text{if assignment } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

J **is the set of all feasible assignments and c_j the benefit from assignment *j***

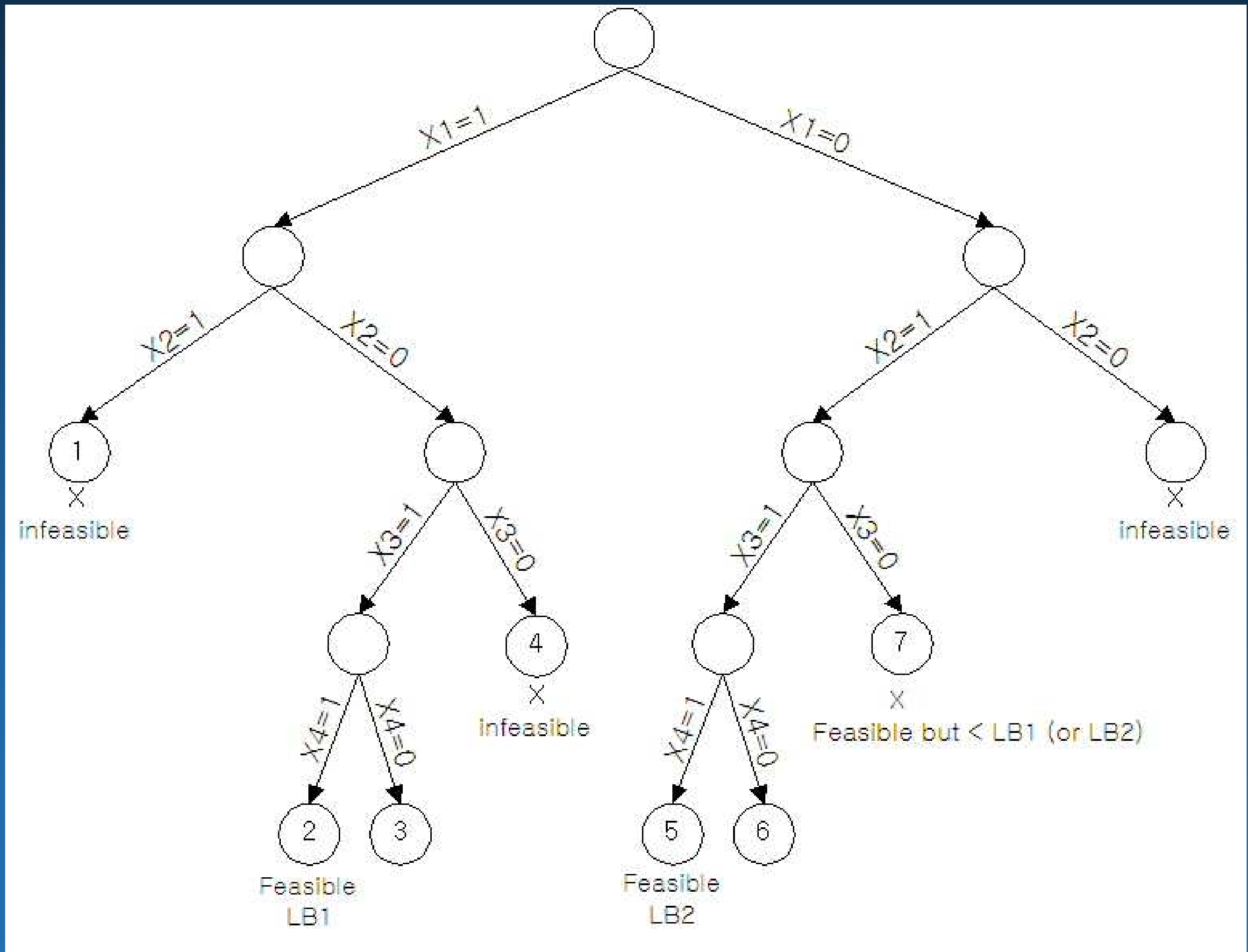
S_s, s = 1,…,m, is the subset of *J* to which source s is assigned

T_t, t = 1,…,n , is the subset of *J* which serves target t

Enumeration Tree Example



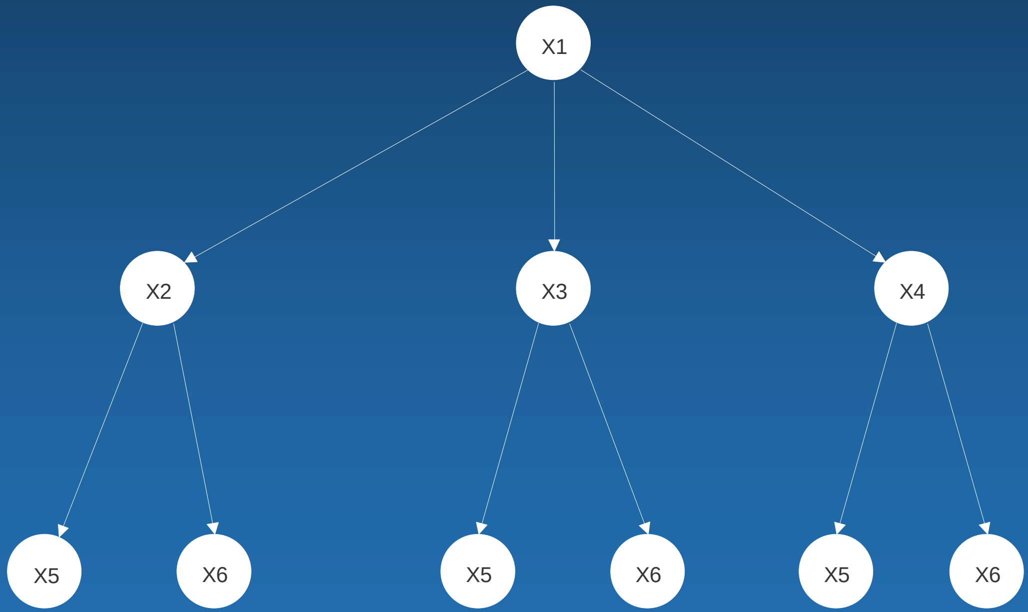
Enumeration Tree Sweep



Branch and Bound - continued

- **Sort Assignments by source in decreasing benefit order**
  - ***Branch on each source***

Source	Assignments
S1	X1
S2	X2 X3 X4
S3	X5 X6



Branch and Bound - continued

- **Incorporate objective tolerance**

  Source conflicts

S1	1	2	3	4	5	6	7
S2	8	9	10	11	12	13	
S3	14	15	16	17	18		

- **If our current optimal solution is at (1,8,15):**

  if

  [total benefit of (1,9,14)] < [total benefit of (1,8,15) – objective tolerance]

  then skip the (1,9,14), (1,9,15), (1,9,16), (1,9,17), (1,9,18) combinations - this will reduce further unnecessary branchings

Branch and Bound - continued

- **Target Conflicts need to be efficiently evaluated**
- **Bitmasks used to characterize targets in an assignment as well as the targets in the current solution**
- **Bitwise AND of current solution target representation (through source levels n) with the target representation for an assignment for source n+1**
- **Single integer for up to 32 targets, use arrays if > 32**

Branch and Bound - continued

- **Will find an assignment with optimal benefit**
- **Large number of enumerations – can be costly**
- **Number of combinations may prove intractable for large m, n**

Sequential Method

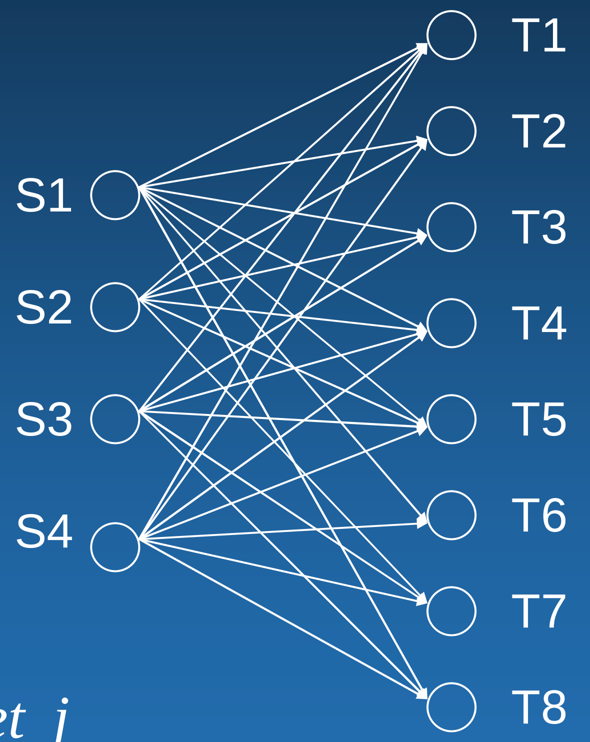
Given m sources and n targets, we set up a directed bipartite graph (G) of sources and targets, and look for the solution of the asymmetric assignment problem

$$\max \sum_{(i,j)\in G} a_{ij} x_{ij}$$

$$s.t. \quad \sum_{\{j|(i,j)\in G\}} x_{ij} \le 1 \quad i = 1,\ldots,m$$

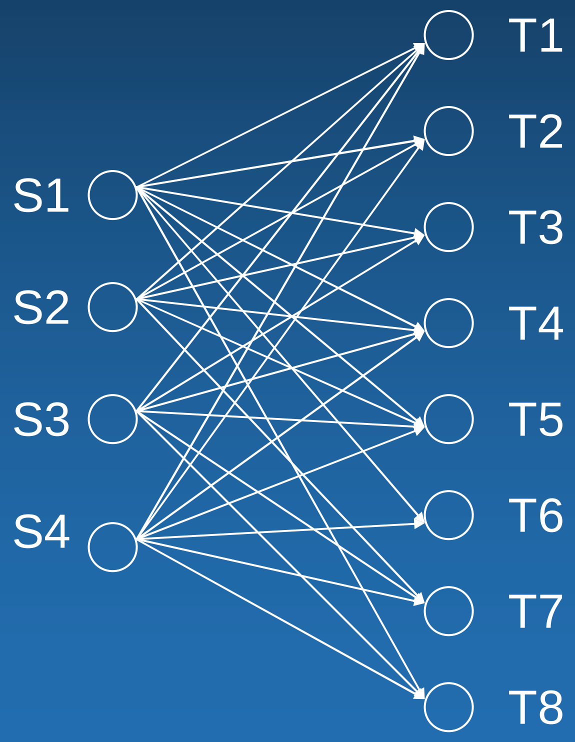
$$\sum_{\{i|(i,j)\in G\}} x_{ij} \le 1 \quad j = 1,\ldots,n$$

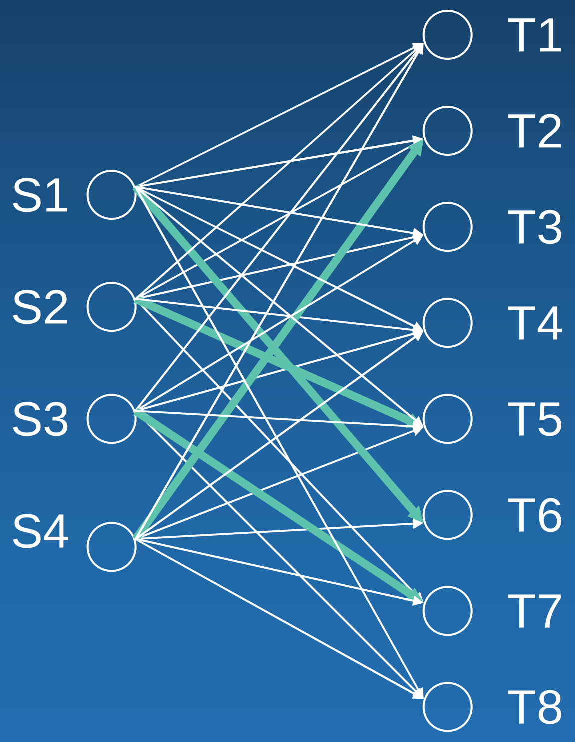
$$x_{ij} = \begin{cases} 1, & if\ \ source\ i\ assigned\ to\ target\ \ j \\ 0, & otherwise \end{cases}$$



Sequential Method – Auction algorithm

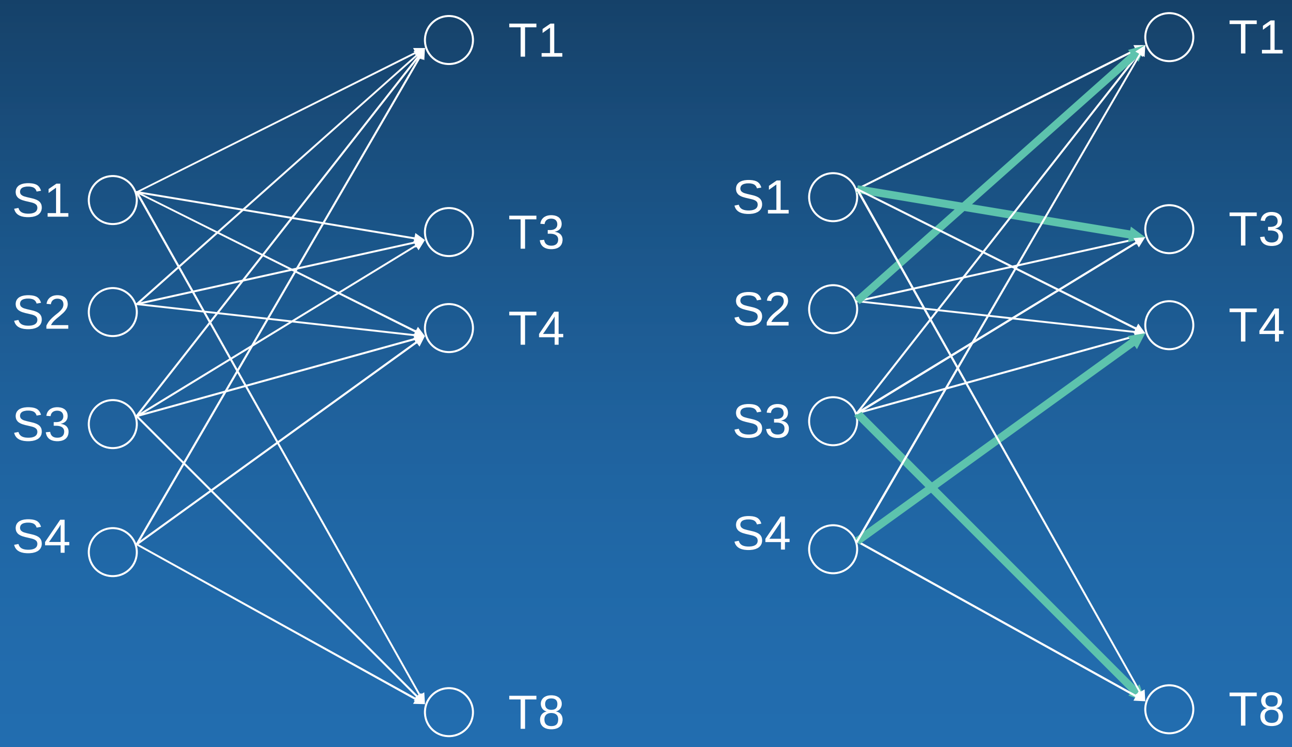
- **Use Bertsekas auction algorithm – multiple passes for m < n**
  - ***First pass***





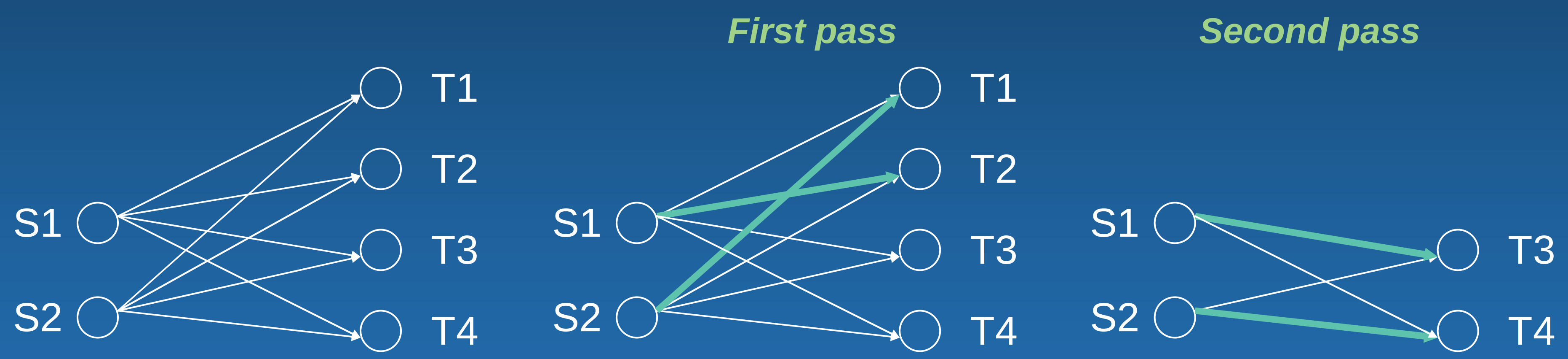
Sequential Method – Auction algorithm

- **Successive passes**
  - *Regenerate graph with remaining resources and targets*



Sequential Method

- Fast, less memory, fewer enumerations
- Greedy approach – may yield less than optimal solution
  - *S1 with Ax2, S2 with Bx2*
  - *T1 (100 ; Ax1, 1.0 or Bx1,1.0), T2 (100 ; Ax1, 1.0 or Bx1,1.0)*
  - *T3 (100 ; Ax1, 0.5 or Bx1,1.0), T4 (100 ; Ax1, 0.5 or Bx1, 1.0)*



Sequential method solution: S1 – T2 – T3, S2 – T1 – T4, Total benefit 350

Branch & bound method solution: S1 – T2 – T1, S2 – T3 – T4, Total benefit 400

Computational Experiments

- Created Monte carlo simulation code for generating different set of sources, targets and their resource requirements, randomly
- Base Parameter Set

Max Targets	2
Max Sources	1
Travel Capacity	25 units
Objective Tolerance	0
Max Time	10
Max Nodes	5000000

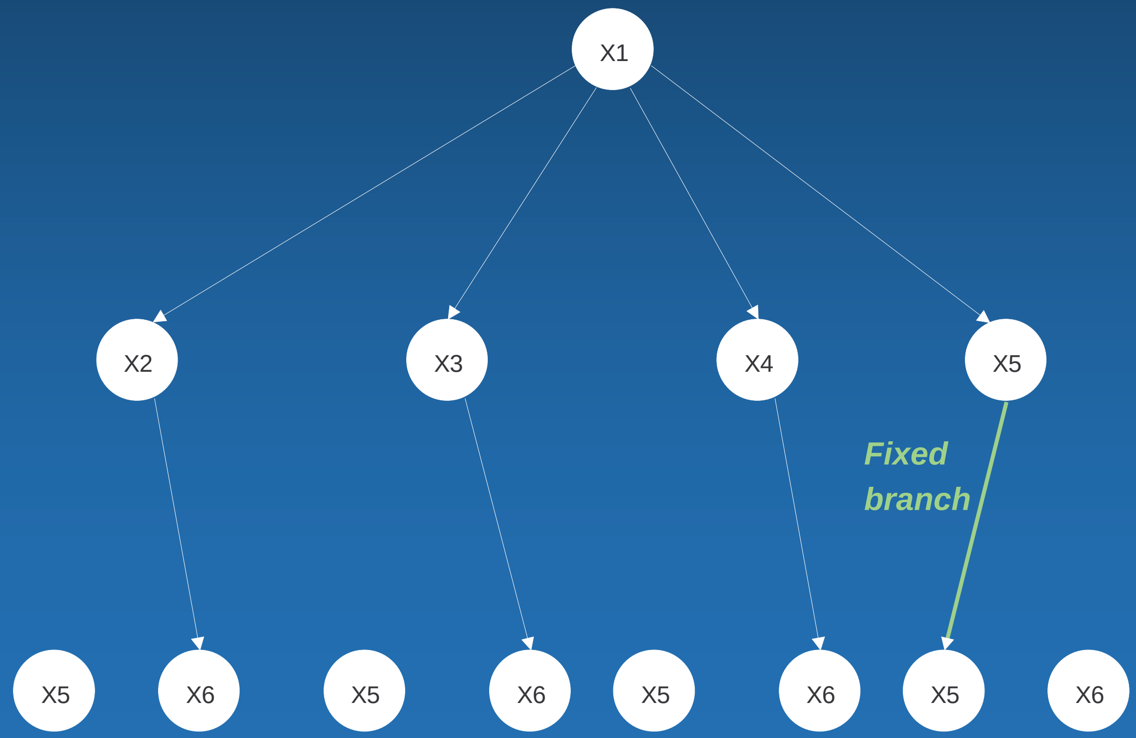
- By modifying one variable in the base every time, we generated different parameter sets
- In each parameter set, we generated 50 instances. Average benefit of those 50 solutions benefit by the Branch and Bound algorithm is compared with the average of the 50 solutions by sequential method
- Sequential method solutions within 1% - 5% of Branch & Bound solutions

Branch & Bound Method – Multiple Targets

- **Allow more than one source per target**
  - ***e.g. Two platforms, each delivering half the required resource for a given effectiveness***

- **Example**

Source	Assignments
S1	X1
S2	X2 X3 X4 X5
S3	X5 X6



Conclusions

- **Investigated suitability of successive auction algorithm for multi-target assignments**
  - ***Branch and Bound serving as benchmark***
  - ***Fast, efficient, fewer enumerations: O(nm) vs. O(nm^2) for 2 targets per source***
  - ***Successive auctions consistently found multiple assignments close to optimal***

- **Gained insight into performance of B&B approach**
  - ***First feasible solution also close to optimal value***
  - ***Differences may be deemed operationally insignificant***
  - ***Can thus provide quick solutions to complicated multi-source assignment problems***

- **Questions?**